Mesh Types for Curvature Regularization

Strandmark, Petter; Kahl, Fredrik

2011

Mesh Types for Curvature Regularization

Petter Strandmark and Fredrik Kahl
Centre for Mathematical Sciences, Lund University
Email: {petter,fredrik}@maths.lth.se

***Abstract*—Length and area regularization are commonplace for inverse problems today. It has however turned out to be much more difficult to incorporate a curvature prior. In this paper we propose two improvements to a recently proposed framework based on global optimization. The mesh geometry is analyzed both from a theoretical and experimental viewpoint and hexagonal meshes are shown to be superior. Our second contribution is that we generalize the framework to handle mean curvature regularization for 3D surface completion and segmentation.**

I. Curvature in Vision

The problem we are interested in solving amounts to minimizing the following energy functional:

$$E(R) = \int_R g(\boldsymbol{x})\, d\boldsymbol{x} + \int_{\partial R} \left(\lambda + \gamma\kappa(\boldsymbol{x})^2\right) dA(\boldsymbol{x}), \quad (1)$$

where R is the 2D (or 3D) foreground region with boundary ∂R. Here $g(\boldsymbol{x})$ is the data term, which may take many forms depending on the application, λ is a positive weighting factor for length (or area) regularization, and γ controls the amount of curvature regularization, denoted κ. Note that the domain may be a 2D image region or a 3D region. In the former case, the boundary is a curve and the notion of curvature is the usual one, while in the latter, the boundary is a surface and κ refers to the mean curvature.

Second order priors like curvature are important for many vision applications, such as stereo [1]. In image segmentation, experiments have shown that curvature regularization is able to capture thin, elongated structures [2], [3] where standard length-based regulators would fail, c.f. Fig 1. Curvature has also been identified as a key factor in human perception based on psychophysical experiments on contour completion [4]. Still, most segmentation-based approaches in computer vision do not use curvature information. This is contrast to length or area regularity which do play an important role. One of the reasons for this fact is that curvature regularity is harder to incorporate in a global optimization framework. Note that curvature regularity is fundamentally different from length or area regularity. While, for example, length regularization prefers shorter boundaries, there is no such bias in curvature regularization. In fact, due to a famous theorem of Werner Fenchel, we know that the integral of the absolute curvature for any closed convex plane curve is equal to 2π.

In differential geometry, energy functionals of the type in (1) have been studied for a long time. In the surface case, the functional is known as the Willmore energy [5]. It gives a quantitative measure of how much a given surface deviates from a round sphere. Local descent techniques have been derived for minimizing (1), cf. [6], but they are very dependent on a good initialization.

(a) Thresholded data term (b) Curvature regularization

Fig. 1: Curvature regularization preserves long, thin objects.

We propose an improvement to the current state-of-the-art of curvature regularization. This results smaller memory requirements, and hence important steps are taken to make curvature regularization more practical. More specifically, we show how to obtain better suited tessellations of the domain. We also show how to apply curvature regularization for surfaces in $\mathcal{R}^3$. To our knowledge, this is the first work that has been able to globally optimize functionals of squared mean curvature.

II. Length and Area Regularization

The basis for our work is the discrete differential geometry framework developed by Sullivan in [7] and Grady in [8] for computing minimal surfaces and shortest paths. The goal is to compute a discrete approximation of the continuous functional in (1). We will recast the problem as an integer linear program and solve it via a linear programming (LP) relaxation. In this section we limit the exposition to the standard case without the curvature term (corresponding to $\gamma = 0$). Interestingly, this integer linear program can be shown to be totally unimodular and hence the LP relaxation will be tight.

The method is based on tessellating the domain of interest into a so-called *cell complex*, a collection of non-overlapping basic regions whose union gives the original domain. Several types of tessellations are possible. Some examples are given in Fig. 2 for 2D and Fig. 3 for 3D meshes. Typical choices are square meshes (2D), resulting in 4-connectivity and cube meshes (3D) giving 6-connectivity. To mimic 8-connectivity, pixels are subdivided into four triangular regions each. We will elaborate more on this issue in Section IV.

The boundaries of 2D regions are called *edges*, and the boundaries of 3D regions are called *facets*. To make this approach work, it is necessary to consider both possible orientations of each edge and facet.

In the integer linear program, there are two sets of binary variables, one reflecting regions and the other related to boundaries. For each basic region, a binary variable reflects whether the region belongs to the foreground or the background. Let x_i, $i = 1, \ldots, m$ denote these binary variables, where m is the number of basic regions. The region integral in (1) is now easily approximated by a linear objective function of the form

$$\sum_{i=1}^{m} g_i x_i. \tag{2}$$

In this paper we let x_i denote region variables and y_i boundary variables (lines in 2D and facets in 3D). The length area term in (1) is then represented with

$$\lambda \sum_i \ell_i y_i, \tag{3}$$

where ℓ_i denotes the length of edge i, and similarly in 3D with areas a_i. To enforce consistency between the region and boundary variables, **surface continuation constraints** [3] are used in the 2D case. We use completely analogous constrains (9) in three dimensions.

When considering surface completion, region variables are not needed and linear constraints of the type (6) may be used to enforce consistency of the surface [8].

III. Curvature Regularization

A. Curvature in the Plane

To be able to handle curvature regularization, *pairs* of boundary variables are introduced. We denote these pairs by y_{ij}. Schoenemann et al. [3] described how to introduce **boundary continuation constraints** to ensure that an actual boundary curve is formed. Without the latter constraint, only straight line pairs would be used.

Having introduced the line pair variables, the last term in (1) may also be represented as a linear function:

$$\gamma \sum_{i,j} b_{ij} y_{ij}. \tag{4}$$

The coefficients b_{ij} used by [9], [3] were

$$b_{ij} = \min\{\ell_i, \ell_j\} \left(\frac{\alpha}{\min\{\ell_i, \ell_j\}} \right)^p, \tag{5}$$

where α is the angle difference between the two lines. We use $p = 2$ exclusively in this paper.

B. Curvature of Surfaces

Each facet in our 3D mesh is associated with a variable $\boldsymbol{y} = (y_1, \ldots, y_{2n})$ of areas $\boldsymbol{a} = (a_1, \ldots, a_{2n})$. There are twice as many variables as facets, because each facet is associated with two variables, one for each orientation. The two are distinguished by (arbitrarily) assigning a normal to each face in the mesh. With the matrix B as defined in [8], the optimization problem for surface completion with area regularization is

$$\begin{aligned} \underset{\boldsymbol{y}}{\text{minimize}} \quad & \lambda \boldsymbol{a}^{\mathrm{T}} \boldsymbol{y} \\ \text{subject to} \quad & \mathsf{B}\boldsymbol{y} = 0; \quad \boldsymbol{y} \in \{0,1\}^{2n}; \quad y_k = 1,\ k \in K. \end{aligned} \tag{6}$$

$$\mathsf{B}_{e,y_i} = \begin{cases} +1, & \text{if edge e borders } y_i \text{ with coherent orientation} \\ -1, & \text{if edge e borders } y_i \text{ with coherent orientation} \\ 0, & \text{otherwise} \end{cases}$$

K is the set of facets that are supposed to be part of the minimal surface a priori. We now extend this formulation to support curvature by introducing face pairs. Each pair of facets in the mesh with an edge in common are associated with two variables $\{y_{ij}\}$ (one for each orientation). Enforcing consistency between the face variables and the variables corresponding to the pairs of faces can be done with linear constraints:

Surface continuation constraints. For each oriented facet k and each one of its edges e we add the following constraint:

$$y_k = \sum_{ij \text{ with edge } e} d_{k,ij} y_{ij}. \tag{7}$$

The sum is over all pairs ij with edge e in common. The indicator $d_{k,ij}$ is 1 if facet k is part of the pair ij.

Having introduced the facet pairs, we follow [10] and associate them with a cost $b_{i,j}$, approximating the mean curvature (compare with (5) on page 2):

$$b_{i,j} = \frac{3||\boldsymbol{e}_{i,j}||^2}{2(a_i + a_j)} \left(2 \cos \frac{\theta_{i,j}}{2} \right)^2, \tag{8}$$

where $\theta_{i,j}$ is the dihedral angle between the two facets in the pair. $||\boldsymbol{e}_{i,j}||$ is the length of their common edge. The objective function we are minimizing is then $\rho \sum_i a_i y_i + \sigma \sum_{i,j} b_{i,j} y_{i,j}$, subject to the constraints in (6) and (7). This approximation is not perfect; for example, it will not give the correct approximation for saddle points. However, it measures how much the surface bends and fulfills a couple of requirements listed by [10].

Segmentation, as opposed to surface completion, require variables for each volume element in order to incorporate the data term. Additional consistency constraints are then required:

Volume continuation constraints. For each facet k,

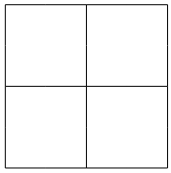
$$\sum_i b_{k,i} y_i + \sum_i g_{k,i} x_i = 0, \tag{9}$$

where b_k indicates whether the facet y_k is positively or negatively incident w.r.t. the chosen face normal. $g_{k,i}$ is 1 if the volume element x_i is positive incident (the face normal points towards its center), -1 if it is negative incident and 0 otherwise. Both sums have two non-zero terms.

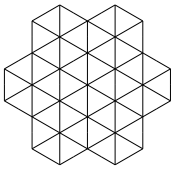
IV. Tessellations

A. Hexagonal Meshes

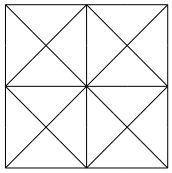
Hexagonal meshes have long been studied for image processing.[11] One characterizing fact of hexagons is that they are the optimal way of subdividing a surface into regions of equal area while minimizing the sum of the boundary lengths.[12] The fact that is more important to us is the neighborhood structure. In a hexagonal lattice every region has 6 equidistant neighbors. When approximating curvature we would like to represent as



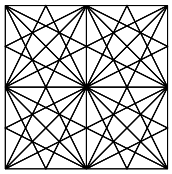
(a) Square mesh with 4-connectivity. Each cell has 1 region and 2 lines (on average).



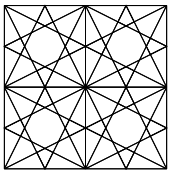
(b) Hexagonal mesh with 6-connectivity. Each cell has 6 regions and 9 lines.



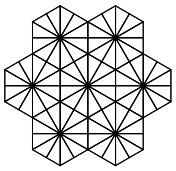
(c) Square mesh with 8-connectivity. Each cell has 4 regions and 6 lines.



(d) Square mesh with 16-connectivity. Each cell has 32 regions and 52 lines.

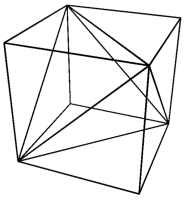


(e) Square mesh with 12-connectivity. Each cell has 25 regions and 44 lines.

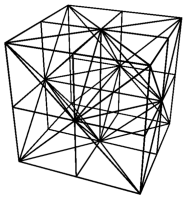


(f) Hexagonal mesh with 12-connectivity. Each cell has 12 regions and 18 lines.

Fig. 2: Different types of grids. The maximum angle between the possible straight lines is 90° in (a), 60° in (b) and 45° in (c). Meshes (d), (e) and (f) have about 27°, 37° and 30° as their maximum angle, respectively.



(a) One unit cube.



(b) Eight unit cubes in a $2 \times 2 \times 2$ mesh.

Fig. 3: ***Interactive figure.*** Each unit cube is split into 5 tetrahedrons. This is the type of mesh used for our experiments in 3D. When stacking several, every other cube has to be mirrored in order to fit.

many different straight lines as possible and we would like the maximum angle between them to be small, as that gives us a better approximation of a smooth curve.[9] The neighborhood structure of the hexagonal mesh allows for similar performance (number of lines and angle between them) while using fewer regions. This is illustrated in Fig. 2, where three crude meshes and three finer meshes are shown. The meshes in Figs. 2d and 2f have similar maximal angle between the possible straight lines, but the hexagonal mesh achieves this with fewer regions due to the favorable intersection pattern of the lines. This suggests that hexagonal meshes can achieve the same accuracy as the meshes (c) and (d) used in [3] with a significantly smaller linear program.

B. Tetrahedrons

The mesh used for the segmentation can be created in a number of ways. [13] The quality of the approximation depend on how many different possible straight lines that can be represented by the mesh, since a larger possible

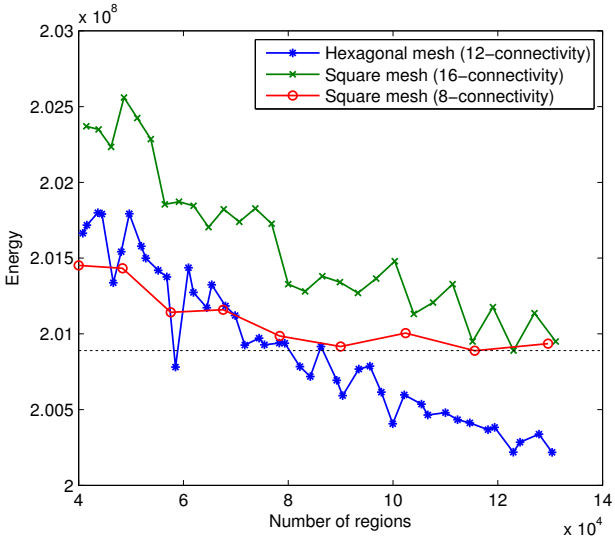


Fig. 4: Optimal energy vs. the total number of regions. The best accuracy obtained by the square mesh was achieved by the hexagonal mesh with about half the number of regions. This experiment used $\lambda = \gamma = 10000$. The energy difference might seem small, but differences of these magnitudes often correspond to significant changes in segmentation.

choice of line slopes allows the mesh to approximate a continuous surface more closely.

There are many choices of tessellations in three dimensions. We have taken a quite simple approach and divided each unit cube into five tetrahedrons, as shown in Fig. 3. This allowed us to be enough different planes to demonstrate the global optimization of mean curvature.

V. Experimental Results

This paper does not focus on how to model the data term and we will use a simple, two-phase version for all our segmentation experiments:

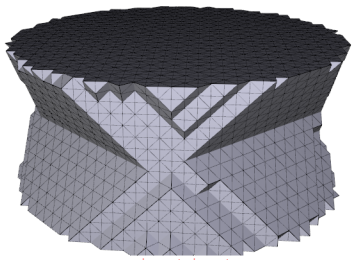
$$g(\boldsymbol{x}) = (I(x) - \mu_1)^2 - (I(x) - \mu_0)^2, \qquad (10)$$

where μ_0 and μ_1 are two fixed mean values and I is the image.

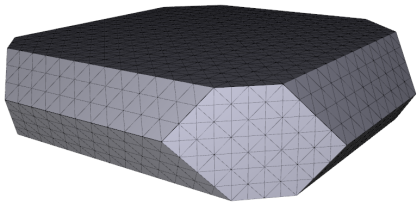
A. Hexagonal Meshes

In our first experiment we evaluate hexagonal vs. square meshes. We are comparing three types of meshes, the 8- and 16-connected square mesh and the 12-connected hexagonal mesh, shown in Fig. 2 (c), (d) and (f). We fixed a data term of a 256×256 image (cameraman) and lay meshes of various types and sizes on top of it and calculated the optimal energy.

The result is shown in Fig. 4, where the optimal energy is plotted as a function of the number of regions used. This is reasonable, since the number of regions is a good indicator of the total size of the linear program. The analogous plots using the number of line pairs or edges look the same. We see that the 8-connected grid converges quickly, but to a suboptimal energy. The hexagonal mesh consistently outperforms the 16-connected grid. If we were to let the number of regions grow very large, the 16-connected grid would probably achieve a lower energy



(a) Area regularization ($40 \times 40 \times 15$ mesh, 447 seconds)



(b) Curvature regularization ($25 \times 25 \times 7$ mesh, 178 seconds)

Fig. 5: ***Interactive figure.*** Surface completion with area and curvature regularization. Two flat, circular surfaces at the top and bottom were fixed. The surface in (a) bends inwards to approximate a catenoid and in (b) it correctly bends outwards to minimize the squared mean curvature.

than the hexagonal, due to it having 2 more possible straight lines. We have not been able to observe this in practice, though, due to the memory requirements.

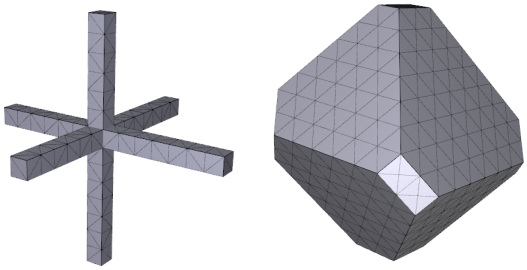
B. The Wilmore Functional

For our experiments in three dimensions we generated a mesh where each unit cube was split into 5 tetrahedrons, see Fig. 3. We then created the set K as two circular surfaces at $z = 0$ and $z = z_{\max}$, with nothing in between. The analytic solution with area penalty is the catenoid, one of the first minimal surfaces found. Fig. 5a show the discrete version obtained with $\lambda = 1$ and $\gamma = 0$. If instead the mean curvature is chosen as the regularizer, the optimal surface instead bends outwards. The solution to this problem is shown in Fig. 5b and is the global optimum, since all variables ended up integral in the LP relaxation of (6). We used Clp as our LP solver.

In another experiment we also used variables for the volume elements. The data term was a 3D 'cross' where the volume elements were forced to be equal to 1, whereas the volume elements at the boundary were forced to be 0. The optimal segmentation when the area was minimized coincided with the data term and is shown in Fig. 6a. When instead minimizing the curvature the optimal segmentation should resemble a sphere, which is observed in Fig. 6b.

VI. Conclusions

We have introduced constraints for 3D surface completion and segmentation. Experiments are encouraging (Figs. 5 and 6) with exclusively globally optimal solutions. To our knowledge, this is the first time the mean curvature



(a) Area regularization (b) Curvature regularization

Fig. 6: ***Interactive figure.*** Surface completion on a $16 \times 16 \times 16$ mesh with area and curvature regularization and volume element variables. The data term and the optimal surface using area regularization coincide. The radius of the volume in (b) is constrained by the mesh size. Otherwise, a minimal surface would not exist for the continuous problem.

of surfaces has been optimized globally. The next step would be to apply this method to e.g. the partial surfaces obtained by stereo estimation algorithms. Another line of further research is how to be able to cope with a finer discretization of the 3D volume.

Acknowledgments

This work has been funded by the Swedish Research Council (grant no. 2007-6476), by the Swedish Foundation for Strategic Research (SSF) through the programme Future Research Leaders, and by the European Research Council (GlobalVision grant no. 209480).

References

[1] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon, "Global stereo reconstruction under second order smoothness priors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2115–2128, 2009.

[2] N. El-Zehiry and L. Grady, "Fast global optimization of curvature," in *Conf. Computer Vision and Pattern Recognition*, 2010.

[3] T. Schoenemann, F. Kahl, and D. Cremers, "Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation," in *Int. Conf. Computer Vision*, 2009.

[4] G. Kanizsa, "Contours without gradients or cognitive contours," *Italian Jour. Psych.*, vol. 1, pp. 93–112, 1971.

[5] T. Willmore, "Note on embedded surfaces," *An. Sti. Univ. "Al. I. Cuza" Iasi Sect. I a Mat. (N.S.)*, pp. 493–496, 1965.

[6] L. Hsu, R. Kusner, and J. Sullivan, "Minimizing the squared mean curvature integral for surfaces in space forms," *Experimental Mathematics*, vol. 1, pp. 191–207, 1992.

[7] J. Sullivan, "Crystalline approximation theorem for hypersurfaces," Ph.D. dissertation, Princeton Univ., 1990.

[8] L. Grady, "Minimal surfaces extend shortest path segmentation methods to 3D," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 321–334, Feb. 2010.

[9] A. M. Bruckstein, A. N. Netravali, and T. J. Richardson, "Epi-convergence of discrete elastica," *Applicable Analysis, Bob Caroll Special Issue*, vol. 79, pp. 137–171, 2001.

[10] M. Wardetzky, M. Bergou, D. Harmon, D. Zorin, and E. Grinspun, "Discrete quadratic curvature energies," *Comput. Aided Geom. Des.*, vol. 24, no. 8-9, pp. 499–518, 2007.

[11] L. Middleton and J. Sivaswamy, *Hexagonal Image Processing: A Practical Approach*. Springer-Verlag New York, Inc., 2005.

[12] T. C. Hales, "The honeycomb conjecture," *Discrete & Computational Geometry*, vol. 25, no. 1, pp. 1–22, 2001.

[13] R. Strand, "Distance functions and image processing on point-lattices," Ph.D. dissertation, Uppsala University, 2008.